Benha Journal of Applied Sciences (BJAS) **print: ISSN 2356–9751**
Vol. (10) Issue (4) (2025), (185- 197) **online: ISSN 2356–976x**
https://bjas.journals.ekb.eg/
engineering sciences

Multi-class Gastrointestinal Diseases improved diagnosis based on Ensemble and Transfer Learning.

Bahaa S. Rabi, Ayman S. Selmy and Wael A. Mohamed
Department of Electrical Engineering, Benha Faculty of Engineering, Benha University, Benha, Egypt.
E-Mail: bahaaalkhtaib@gmail.com

Abstract

From the mouth to the anus, the digestive system is a long tube made up of multiple hollow organs. To heal, people with gastrointestinal illnesses need to receive the correct therapy and be diagnosed as soon as possible. In biomedical applications, there has been a recent surge in research on classifying endoscopic images for the identification of gastrointestinal tract disorders. Deep learning algorithms, particularly Deep Convolutional Neural Networks, are utilized to diagnose major gastrointestinal conditions like ulcerative colitis, polyps, and esophagitis. Multiple approaches were utilized to enhance the performance of the diagnosis system, but they are still not enough due to a lack of datasets and the complexity of designing new algorithms. So, to achieve this goal and overcome these problems, in this proposed work, at first data was preprocessed in manner training and testing, and data augmentation was applied to maximize training operation based on more data, then are four different experiments, the first two experiments were based on two novel different Convolution Neural Network models, which yielded accuracy values of **0.75125** and **0.99875**, respectively; ensemble learning was used in the third experiment, yielding a total accuracy of **0.995**, and transfer learning, which used the well-known pretrained model (**VGG16**), produced an accuracy of **0.9800** in the final experiment. Based on these four experimental methods, results have better performance evaluation parameters in accuracy, specificity, and F1 score than other recent related approaches for gastrointestinal diseases multi-class diagnosis. Also, these techniques can be applied to other multi-class diseases.

Keywords: Computer-Aided Diagnosis (CAD), Deep Convolution Neural Network (DCNN), Deep Learning (DL), Gastrointestinal Diseases Classification, Gastrointestinal Diseases Diagnosis, Ensemble Learning (EL), Transfer Learning (TL), VGG16.

1. Introduction

A wide range of abnormal outcomes, from minor pain to life-threatening diseases, can result from diseases of the human digestive system [1].The International Agency for Research on Cancer has recently predicted that there will be approximately 4.8 million cases of gastrointestinal cancer worldwide, accounting for roughly 26% of all cancer cases and 35% of all cancer-related deaths [2]. Early diagnosis decreases the risk of death and enables the efficient treatment of several GIDs.

The primary drawback is that many digestive disorders go undetected or are confused with other medical professionals when they are screened for them because of noise in the images that conceals significant characteristics [3].

The visual assessment of endoscopic images is subjective, difficult, and rarely repeatable, which raises the possibility of a false diagnosis, according to reports.

In various gastrointestinal endoscopic applications, artificial intelligence (AI) has the potential to enhance clinical practice and improve the efficacy and precision of current diagnostic techniques.

Several machine learning (ML) methods, including transfer learning (TL), ensemble learning (EL), and deep learning (DL), have been employed in gastrointestinal endoscopy.

The DL model, a more complex kind of artificial neural network with several layers, either linear or non-linear, is used in this study.

The layers of a DL model are connected to the layers above and below them by means of particular weights [4]. Text, music, pictures, and numbers are just a few of the data sources from which DL models may effectively extract hierarchical features.

The effectiveness of these DL models allows them to address problems in semi-supervised and unsupervised learning, regression, and recognition.

The goal of EL is to integrate data fusion, data modelling, and data mining into a unified framework. Specifically, a number of EL transformations are used to initially extract a set of features [5].

A type of ML called TL uses knowledge from a single task or dataset to enhance model performance on a related task or dataset [6].In this study, the GIDs dataset has high similarity between different classes. The results section presents some differences between the two deep convolutional neural networks (DCNNs), based on tuning every DCNN model, besides employing the EL technique to overcome the weakness of the DCNN1 model. Finally, applying the TL method to investigate its effectiveness in improving the diagnosis of gastrointestinal diseases (GIDs) based on a pre-trained model (VGG16). All previous techniques have different results, but at last, that means they can be employed as a GID's diagnosis more accurately. Then

the specialist doctors who deal with GIDs patients can follow the correct protocol for curing.

The study's subsequent sections are as follows: Section 2 of this paper introduces the relevant work. The methodologies and a detailed description of the proposed study are presented in Section 3. The evaluation metrics and the GIDs image dataset are described in Section 4. Section 5 presents and discusses the findings. Finally, the final section highlights conclusions and future study.

2. Related Work

To support the accurate diagnosis of various GIDs by clinics. Endoscopic techniques like colonoscopy, esophageal-gastroduodenoscopy, and capsule endoscopy have been investigated for the possibilities of DL algorithms during the past few decades [2].
These days, DL techniques—particularly DCNNs have developed into strong machine learning approaches for image processing applications like GIDs classification.

Multiple techniques have been employed in a number of studies to identify GIDs.
This literature study mainly concentrated on earlier studies that employed the Kvasir dataset [7] and DCNNs for classification in relation to earlier work on GIDs classification.

- The authors of this study [8], Mousa Alhajlah, Muhammad Nouman Noor, Muhammad Nazir, Awais Mahmood, Imran Ashraf and Tehmina Karamat, suggested a feature extraction method based on the Mask Recurrent-Convolutional Neural Network (R-CNN) and optimized pre-trained ResNet-50 and ResNet-152 networks. R-CNN is first used to identify the region of interest, and then it is used to train improved models through transfer learning. A serial technique is utilized to fuse the fine-tuned models once features have been extracted. Additionally, the optimal feature selection from the fused feature vector has also been chosen by an Improved Ant Colony Optimization (ACO) algorithm. Finally, machine learning algorithms are used to classify the best-selected characteristics. Using the publicly accessible dataset, the experimental procedure produced an enhanced accuracy of 96.43%.

- Using the dataset gathered from 854 patients, the authors of [9] developed ML model using logistic regression (LR) and ridge regression (RR), which yielded accuracy rates of 82.6% and 83.3%, respectively.

- The authors, M. H. Al-Adhaileh, E. M. Senan, F. W. Alsaade, T. H. H. Aldhyani and N. Alsharif [10], employed kvasir [7] pictures that were preprocessed for improvement and noise reduction in the study before being trained by three neural networks: AlexNet, ResNet-101, and Google Net. This method's greatest accuracy was 97%.
- To enhance the prediction of GI tract abnormalities, the authors S. Nadeem, M. Atif Tahir, S. Sadiq Ali Naqvi, and M. Zaid [11] developed a novel ensemble approach based on DL that combines deep features. Additionally, ML and multimedia content analysis were studied. The testing samples have an F1 score of 0.821 and an accuracy of 83% when logistic regression and the ensemble of various extracted features are used.
- The authors A. Asperti and C. Mastronardo1 [12] demonstrated that data augmentation offers a legitimate remedy for the predetermined data set's limited dimension. They verified that deep learning algorithms can effectively tackle the issue of automatically diagnosing gastrointestinal disorders compared to other methods. Classification is improved by using TL [13], CNN [14], Data Augmentation (DA) Techniques [15], and Snapshot Ensemble [16]. The model is particularly good at categorizing samples that belong to the normal pylorus and normal cecum, according to the authors' findings. Dye-lifted polyps and dyed resection margins are the most common misclassifications. According to the authors, the two classes consist of photos that are extremely comparable in terms of their blue content. Additionally, a few other cases that were misclassified are associated with esophagitis and a normal z-line. By using the new larger edition of the data set or letting the network train more often, miss-classifications might be prevented. Increasing the number of samples from the z-line and esophagitis classes, as well as from the dyed lifted polyps and colored resection margins, may help to improve prediction accuracy. An ensemble of inspection, fine-tuning, and data augmentation is the technique used. The accuracy of this approach was 0.915.
- The TL technique was the main emphasis of the work by authors T. Agrawal, R. Gupta, S. Sahu, and C.E. Wilson [17]. To acquire representations of the medical images, they employed pre-trained VGGNet and networks of Inception-V3. They also extracted a number of properties from these CNNs.The authors employed various characteristic combinations to predict three different classification structures. The applied approach made advantage of the Inception-V3 and VGGNet features from kvasir [7]. With an accuracy of 0.961, the findings show an F1-score of 0.847 and an MCC of 0.826. All of the aforementioned connected issues offered excellent alternatives for categorizing the GIDs, but in this study, two new deep learning algorithms (DCNN1, DCNN2) were applied to compete with the related previous approaches in performance and resource consumption; the study methods will be described in the following section.

3. Methods

Four methods for classifying GIDs utilizing DL, EL, and TL are presented in this paper. Two methods that are based on the DCNN methodology are used in DL.
The procedure involved 4 primary steps are data preparation and feature extraction, and classification

using two new models (**DCNN1** and **DCNN2**). Additionally, two other strategies based on EL and TL techniques are used.

The **DCNN1, DCNN2** architecture, EL, TL, and DA techniques are explained in this section, as well as the details of the recommended methodologies.

3.1 The CNN Main Components

Convolution layers (Conv_Lay), pooling layers, and Fully Connected (FC) layers compose the DCNN network, a unique type of DL strategy.

One of Conv_Lay's benefits is its capacity to extract the entire feature set of the image.

To reduce the size of the image features and streamline network computation, the pooling layer compresses the input feature picture [4].

In order to compress the two-dimensional visual qualities into a one-dimensional feature vector, the FC layer is placed behind the Conv_Layer. Figure (1) defines the Basic **DCNN** architecture.

The **DCNN** has two main procedures for training: first, forward propagation, and second is back propagation.

In the next equations, it will be explained.

- **In the forward propagation**

convolution layer, the function is defined as follows:

$$\chi_j^{\ell}=f\left(\sum_{i\in M_j}\left(\chi_i^{\ell-1}\otimes k_{ij}^{\ell}+b_j^{\ell}\right)\right) \qquad (1)$$

χ_j^{ℓ} indicates the g characteristic of the ℓ Layer image. Equation (1) displays a feature map with all the associations for the $(\ell-1)$ Layer on the right side. M_g Defines the total of sub-matrices. $\chi_i^{\ell-1}$ and the jth convolution kernel $k_{ij}^{\ell}$of the ℓth Layer is convoluted and summed with an offset term. b_j^{ℓ} [18].

a) An activation function (Act_Func) Relu of f(x).

b) In the pooling layer, the forward propagation function is defined in [18]:

- **The backpropagation** rules for DCNN are defined by this equation (2).

$$C(w,b)\equiv\frac{1}{2n}\sum_x\|y(x)-a\|^2 \qquad (2)$$

Where 'w' represents the collection of weights, 'b' is all the biases, 'n' is the number of training input data, and 'a' is the desired output vector.

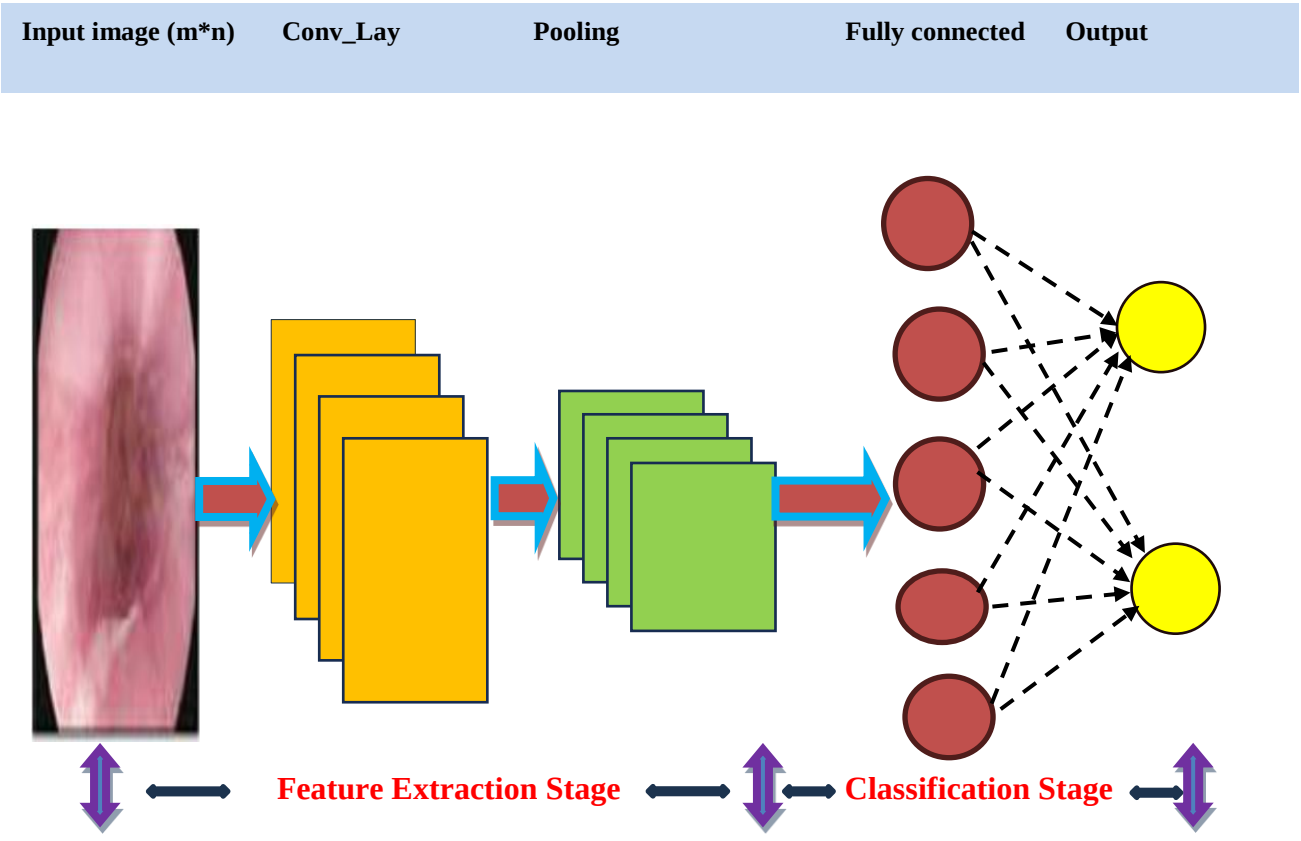


Fig. (1) Basic DCNN architecture.

3.1.1 Activation Function

Act_Func provides the **DCNNs** with non-linearity [4]. Since many real-world events display complex, nonlinear behaviour, this nonlinearity is significant. A model that just uses linear processes is insufficient to describe these events. Act_Func allows for the modelling of a greater range of functions and enhances **CNN** expressiveness. To add non-linearity to the network, **CNNs** use a range of activation functions. One often used component is Act_Func.

- Rectified Linear Unit (Relu)
- Sigmoid function
- Hyperbolic Tangent (Tanh).

• Due to its simplicity and efficiency in addressing the vanishing gradient issue during training, **Relu** is a very commonly used activation function in contemporary CNNs. Conv_Lay is followed by Relu in the standard AlexNet. Relu's mathematical expression is displayed in equation (3).

$$f(x) = \text{Max}(0, x) \quad (3)$$

where x represents the neuron's input.

• The **Sigmoid** function is a great choice for binary classification jobs since it provides an S-shaped mapping from 0 to 1. The mathematical form of a sigmoid is represented by equation (4).

$$f(x)=1/(1+e^{x}) \quad (4)$$

Where x represents the neuron's input.

• The hyperbolic tangent (**Tanh**) function converts each real integer into a value between -1 and 1, much like the sigmoid function does. It is also a non-linear function that contributes to the model's non-linearity. Tanh's mathematical formulation is found in equation (5).

$$f(x)=\frac{(e^{x}-e^{-x})}{(e^{x}+e^{-x})} \quad (5)$$

Where x is the input to the neuron.

3.1.2 Optimization Algorithms

Optimization strategies are used to adjust the weights and biases of a neural network during training to lower the loss function [19]. Among the optimization methods are Adam, AdaGrad, momentum, and stochastic gradient descent (SGD). The Adam was used in this work, and its formula is as follows:

- **Adam**: This technique creates a customizable learning rate that works well for various neural networks by combining ideas from AdaGrad and momentum. The mathematical formulas [20] represent Adam's algorithm (6–10);

$$m_t=\beta_1 m_{t-1}+(1-\beta_1)\nabla_W L(W_{t-1}) \quad (6)$$

$$\Lambda_t = \beta_2 \Lambda_{t-1} + (1-\beta_2)(\nabla_W L(W_{t-1}))^2 \quad (7)$$

$$\widehat{m}_t = \frac{m_t}{1-\beta_1{}^t} \quad (8)$$

$$\widehat{\Lambda}_t = \frac{\Lambda_t}{1-\beta_1{}^t} \quad (9)$$

$$W_t = (W_{t-1}) - \frac{\alpha}{\sqrt{\hat{v}_t+\epsilon}}\widehat{m}_t \quad (10)$$

Λ_t is the momentum at time t, β is the momentum coefficient, $\nabla_W L(W_{t-1})$is the gradient of the Loss_Func concerning the weights at the time $t-1$, α is the learning rate, and W_t is the updated weights.

3.2 Data Augmentation

Training new ML models is the main application for data augmentation (DA), which is the act of creating new data artificially from preexisting data. Large and varied datasets are necessary for the initial training of ML models; however, finding sufficiently diverse real-world datasets can be difficult due to data silos, legal restrictions, and other issues. The technique of making minor adjustments to the original data to artificially enlarge the dataset is known as DA. For rapid and high-quality DA, generative AI technologies are being used more and more in several industries [21].

3.3 Ensemble Learning

Traditional ML techniques may not produce adequate results when handling complex data, such as imbalanced, high-dimensional, noisy data, etc., despite notable advancements in knowledge discovery. This is because these approaches have trouble capturing the various features and underlying structure of the data [22].

This makes the question of how to build an effective information discovery and mining model a crucial one in the data mining discipline. One area of intense research is ensemble learning (EL), which attempts to combine data mining, data modelling, and data fusion into a single framework. In particular, a collection of features is first extracted using a range of transformations in EL. Multiple learning algorithms are used to generate weak prediction outcomes based on these learned properties. Lastly, EL combines the useful information from the aforementioned findings to improve predicted performance and knowledge discovery through adaptive voting schemes [22].

Several ML algorithms are used in EL techniques, and one of them generates poor predictive results based on features gleaned from a variety of data projections. The results are then combined with various voting mechanisms to achieve better performance than any single algorithm could [23]. Figure (2) illustrates how the learning model's overall error consistently decreases until it reaches the bottom, after which it rapidly rises as the model's complexity rises.

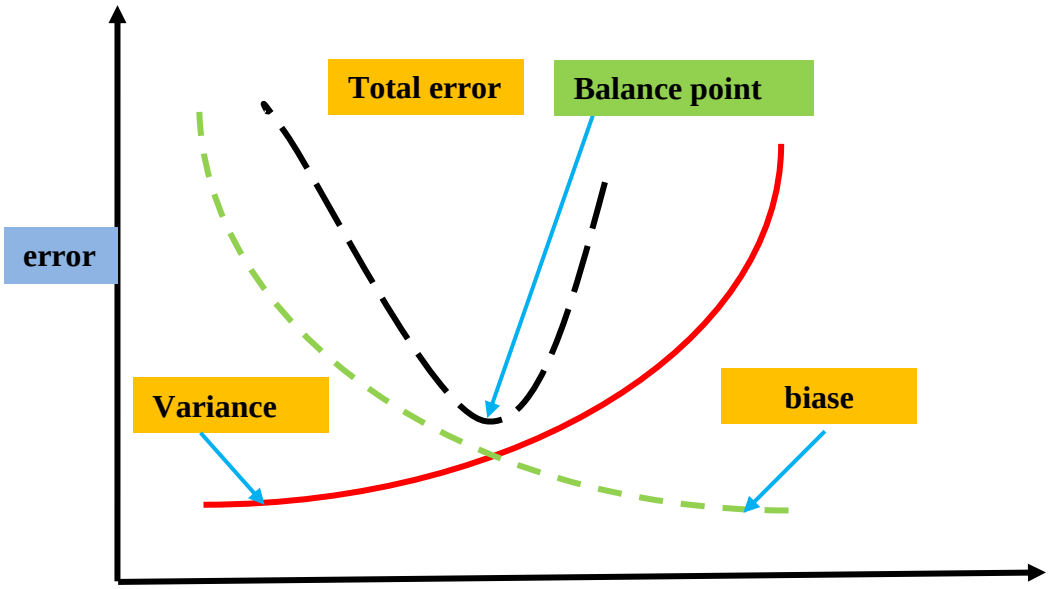


Fig (2) The relationship between learning curve and model complexity.

It is evident that the learning model's total error consistently decreases until it reaches the bottom, after which it rapidly rises as the model complexity rises. The variance and bias have opposing changes: the variance remains steady before rising sharply, while the bias drops sharply before remaining steady.

3.4 Transfer Learning Overview

Training and testing data in traditional ML utilize the identical input feature space and data distribution. The performance of a predictive learner may suffer when the distribution of data between the test and training sets differs [24,25].

It might be challenging and costly in some situations to find training data that matches the feature space and expected data distribution properties of the test data.

As a result, a high-performance learner for a target domain that has been taught from a similar source domain must be developed. This becomes the driving force for TL.

The distinction between the learning procedures of traditional and transfer learning approaches is depicted in Figure (3).

As we can see, transfer learning techniques aim to apply the knowledge from some prior tasks to a target task when the latter has fewer high-quality training data. In contrast, typical machine learning techniques aim to learn each task from scratch.

By using knowledge from a similar area, TL helps learners from one domain become better. It can be discovered why TL is possible by using non-technical real-world experiences.

Take two individuals who wish to learn how to play the piano as an example. One individual has never played

music before, whereas the other has played the guitar for

a long time and knows a lot about it.

By applying their prior understanding of music to the job of learning to play the piano, someone with a large background in music will be able to pick up the instrument more quickly [26,27]. A person can learn a related task by using the knowledge they have gained from previously completed work.

The task of predicting the text sentiment of product reviews, when there is an abundance of labelled data from digital camera reviews, is a specific example from the field of machine learning.

To obtain good prediction results, conventional machine learning approaches are applied if both the training and target data are taken from digital camera reviews. However, because the target data is from food reviews and the training data is from digital camera reviews, the prediction results are likely to suffer from the disparity in-domain data.

There are still many similarities, if not precise ones, between reviews of digital cameras and reviews of meals.

They both express opinions regarding purchased goods and are written in textual form using the same terminology

.

TL can enhance a target learner's performance because of the connection between these two domains [27].

In a TL context, the training and target data can be viewed as existing in distinct sub-domains connected by a high-level common domain.

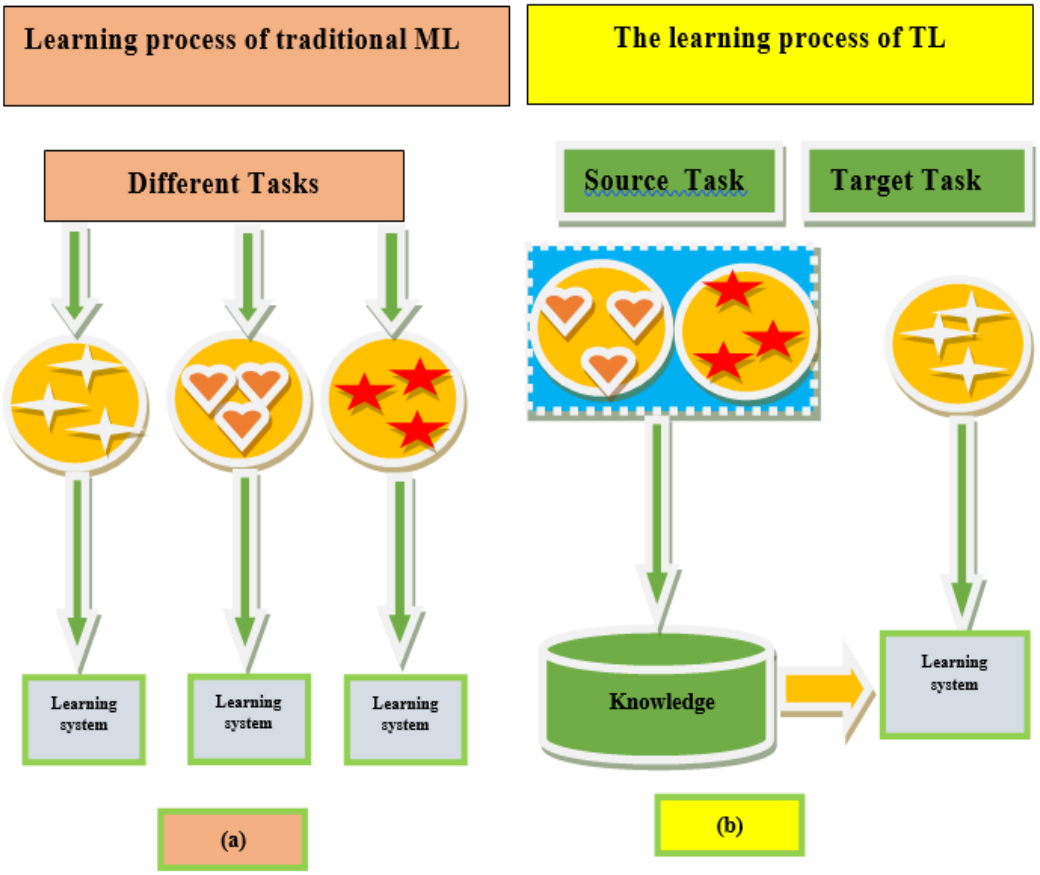


Fig. (3) Different learning processes between (a) traditional machine learning and (b) transfer learning.

3.5 Proposed DCNN1

This DCNN1 comprises three Conv-lay, two max-pool, and two fully connected.

A Relu activation function is applied for all layers except the last fully connected layer, which is SoftMax. As shown in Figure (4), the DCNN1 architecture is plotted. The full DCNN1 architecture is explained as follows:

- **[227x227x3] INPUT**
- **[225x225x32] Conv1: 32 3x3 filters at stride 1, pad 0, Relu.**
- **[112x112x32] MAX POOL1: 2x2 filters at stride 2, pad (valid)**
- **[110x110x64] Conv2: 64 3x3 filters at stride 1, pad 0, Relu.**
- **[55x55x64] MAX POOL2: 2x2 filters at stride 2**
- **[53x53x128] Conv3: 128 3x3 filters at stride 1, pad 0, Relu.**
- **[26x26x128] MAX POOL3: 2x2 filters at stride 2, pad (valid)**
- **[128] FC4: 128 neurons, Relu.**
- **[8] FC5: 8 neurons (class scores).**

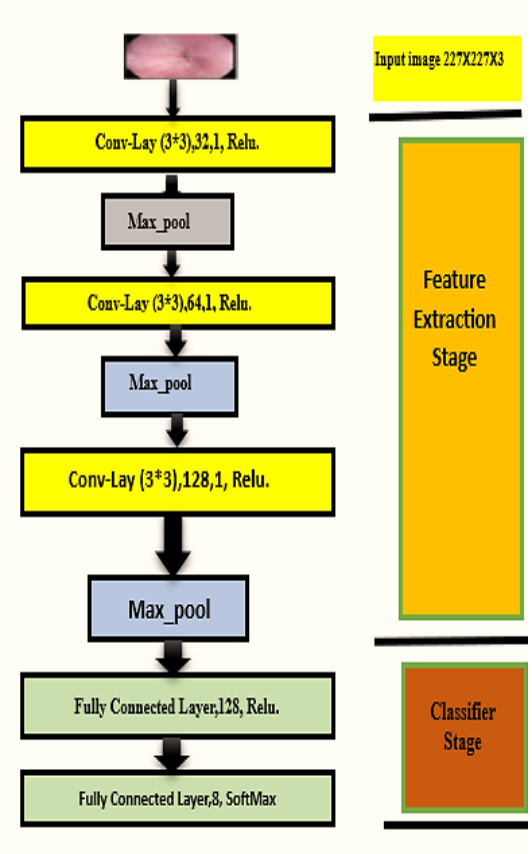


Fig (4) The architecture of the DCNN1 model

3.6 Proposed DCNN2

This DCNN2 comprises five Conv-lay, five max-pool, and two fully connected. A Relu activation function is applied for all layers except the last fully connected layer, which is SoftMax.
As shown in Figure (5), the DCNN2 architecture is plotted. The full DCNN2 architecture is explained as follows:

1) [227x227x3] INPUT.

2) Five Conv_Lay, all of them apply 5x5 filters, no of filters from first to five is (16,32,64,128,512).

3) Five MAX POOL, all of them apply 2x2.

4) Two fully connected layers.

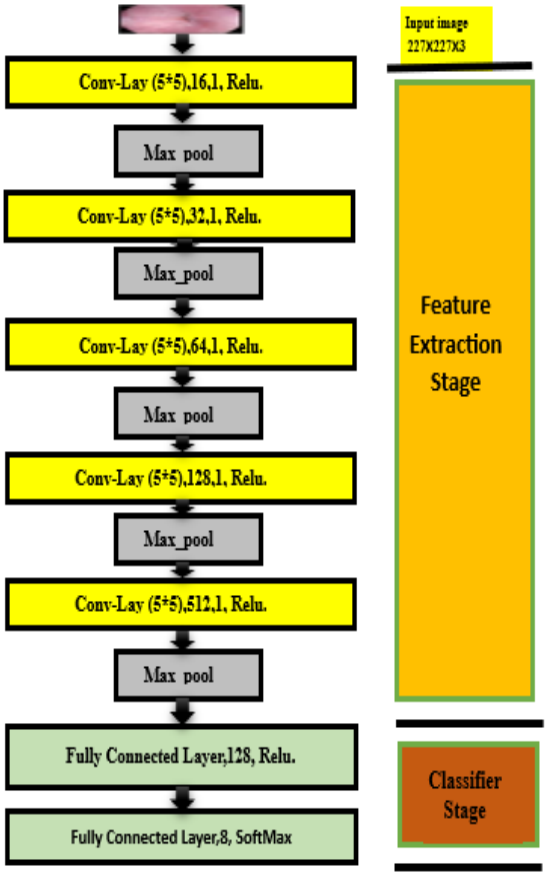
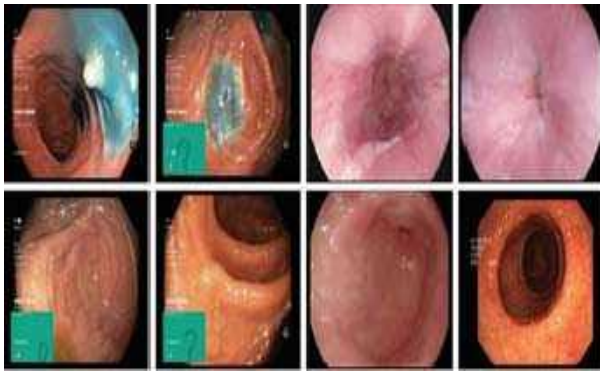


Fig (5) The architecture of the DCNN2 model.

4. Performance Evolution

4.1 Dataset Description

The dataset employed in this work is from research released under the title KVASIR [7]. It's 4,000 images that display eight different multi-class GI situations. Keeping in mind the names of each case, the pictures have been separated into distinct folders. They are between 720x576 and 1920x1072 pixels in size. Some of the offered picture classes include a green image that displays the position and shape of the endoscope inside the gut. Although it may be crucial for further research, this material is given even if it must be used carefully to uncover the endoscopic findings. Figure (6) shows a variety of image examples of different GI illnesses.



Fig(6). From bottom right to top left, the GID photo samples of the KASVIR dataset are: Ulcerative Colitis, Normal Pylorus, Polyps, Normal Cecum, Normal Z-line, Esophagitis, Dyed Resection Margin, and Dyed Lifted Polyp.

4.2 Evaluation Metrics

A confusion matrix, a commonly used assessment approach in image categorization, can be utilized to assess the learning performance of the DCNN model.
The confusion matrix allows us to distinguish between precise and imprecise predictions and helps quantify the degree to which the expected results agree with the actual values.
It also sheds light on the particular kinds of mistakes the model makes. A collection of test and validation data, with the values of the acquired results, is needed to calculate the confusion matrix.
To determine the key parameters, such as accuracy and sensitivity, etc., utilize the confusion matrix.

Recall (**Rec**): Recall (Rec): Another common name for this measure is sensitivity.

$$\textbf{Sensitivity}=TP/(TP+FN) \quad (11)$$

Precision (**Perc**): Another common name for this measure is the positive predictive value.

$$\textbf{Precision}=TP/(TP+FP) \quad (12)$$

Accuracy (**Acc**): The proportion of class labels that were correctly identified:

$$\textbf{Accuracy}=TP+TN/(TP+TN+FP+FN) \quad (13)$$

F1 score: Determined by taking the harmonic mean of the precision and recall, assessing an exam's accuracy:

$$\textbf{F 1 score}=2TP/(2TP+FP+FN) \quad (14)$$

where the notion of "true positive" (**TP**) refers to the number of positive samples that are correctly detected.
The true negative (**TN**) is the number of correctly detected negative samples. False Positives (**FP**) are the number of samples that were incorrectly identified as positive. False Negative (**FN**) refers to the number of samples that are incorrectly labelled as negative.

5. Results and Discussion

This section addresses the outcomes of the four proposed approaches. The eight GIDs classes in the study were then identified as follows:
dyed-Lifted Polyps (0), Dyed-Resection-Margins (1), Esophagitis (2), normal-Cecum (3), normal-Pylorus (4), normal Zline (5), Polyps (6) and Ulcerative colitis (7).
The dataset was divided into two groups for training and testing operations, with a ratio of 80%:20%. The photos in the eight classes were resized to 227□227 pixels. There were then eight classes, with
3200 images for training and 800 images for validation. In addition, the following four approaches used data augmentation.
Hardware resources such as a CPU core i5, 16 gigabytes of RAM, and an Intel GPU were used to process the model. The Spyder compiler on the Anaconda platform uses Python 3.7 as the programming language and Windows 10 as the operating system for software resources. The outcomes are noted as follows.

5.1 $\mathbf{DCNN_1}$ 1st Approach

The results are recorded in this experimental approach $\mathbf{DCNN_1}$, training, to classify the GIDs according to the next hyperparameters, the batch size for training was equal to 128, the number of epochs was equal to 100, the optimizer employed was SGD, 3 dropout(0.2), and the last $\mathbf{DCNN_1}$ A fully connected layer was adapted to 8 because the classes in the dataset are 8 classes.
Table (1) represents a summary of $\mathbf{DCNN_1}$ and total hyperparameters, which are equal to 11169992, occupying 42.61 MB. After training, the accuracy curve, loss curve, and confusion matrix are gathered and plotted in Figures (7), (8), and (9). As shown, the accuracy for testing is about **0.75125**, and the time complexity is 3 hours,13 minutes, and 12 seconds.

Table (1) The $\mathbf{DCNN_1}$ Summary and total parameters

The Layer	Layer Output Shape	No of Parameters
conv2d#1 (Conv2D)	(None, 225, 225, 32)	896
max_pooling2d	(None, 112, 112, 32)	0
drop#1 (Dropout)	(None, 112, 112, 32)	0
conv2d#2_(Conv2D)	(None, 110, 110, 64)	18496
max_pooling2d	(None, 55, 55, 64)	0
drop#2 (Dropout)	(None, 55, 55, 64)	0
conv2d#3(Conv2D)	(None, 53, 53, 128)	73856
max_pooling2d	(None, 26, 26, 128)	0
drop#3 (Dropout)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense#1 (Dense)	(None, 128)	11075712
dense#2 (Dense)	(None, 8)	1032
Total parameters		**11169992 (42.61 MB)**

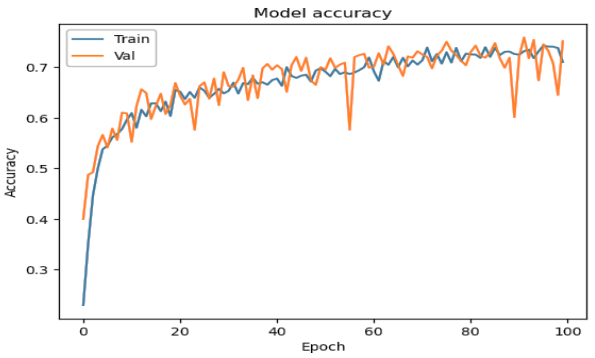


Fig (7) The Accuracy curve of the training and validation curve for the $\mathbf{DCNN_1}$ model.

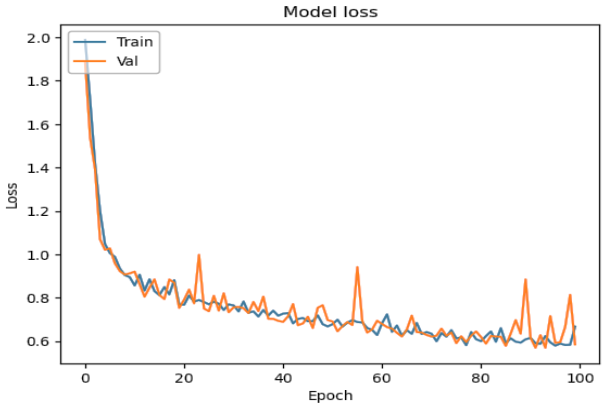


Fig (8) The Loss curve of training and validation curve for the $\mathbf{DCNN_1}$ model.

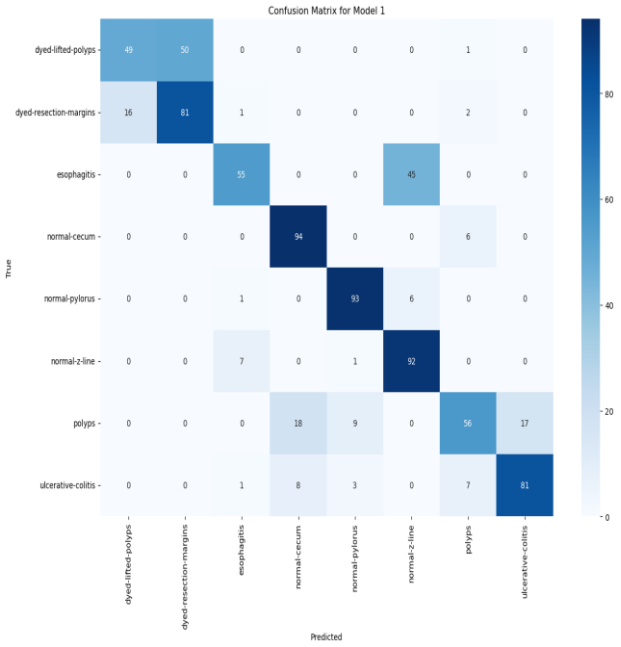


Fig (9) Confusion matrix for $\mathbf{DCNN_1}$ model.

5.2 $\mathbf{DCNN_2}$ 2nd Approach

The results in this experimental approach $\mathbf{DCNN_2}$, training, to classify the GIDs according to the next hyperparameters, the batch size for training was equal to 128 batch size, the number of epochs was equal to 200, the optimizer employed was Adam, 1 dropout (0.2), and the last AlexNet fully connected layer was adapted at 8 because the classes in the dataset are 8 classes. Table (2) represents a summary of $\mathbf{DCNN_2}$ and the total hyperparameters are the same $\mathbf{DCNN_2}$, which

is equal to 4273064, occupying 16.30 MB. After training, the accuracy curve, loss curve, and confusion matrix are gathered and plotted in Figures (10), (11), and (12). As shown, the accuracy for testing is about **0.99875**, and the time complexity is 3 hours,56 minutes, and 1 second.

Table (2) The $\mathbf{DCNN_2}$ Summary and total parameters

The Layer	Layer Output Shape	No of Parameters
conv2d#1 (Conv2D)	(None, 223, 223, 16)	1216
max_pooling2d	(None, 111, 111, 16)	0
conv2d#2_(Conv2D)	(None, 107, 107, 32)	12832
max_pooling2d	(None, 53, 53, 32)	0
conv2d#3(Conv2D)	(None, 49, 49, 64)	51264
max_pooling2d	(None, 24, 24, 64)	0
conv2d#4(Conv2D)	(None, 20, 20, 128)	204928
max_pooling2d	(None, 10, 10, 128)	0
conv2d#5(Conv2D)	(None, 6, 6, 512)	1638912
max_pooling2d	(None, 3, 3, 512)	0
flatten (Flatten)	(None, 4608)	0
dense#1 (Dense)	(None, 512)	2359808
drop#1(Dropout)	(None, 512)	0
dense#2 (Dense)	(None, 8)	4104
Total parameters	**4273064 (16.30 MB)**	

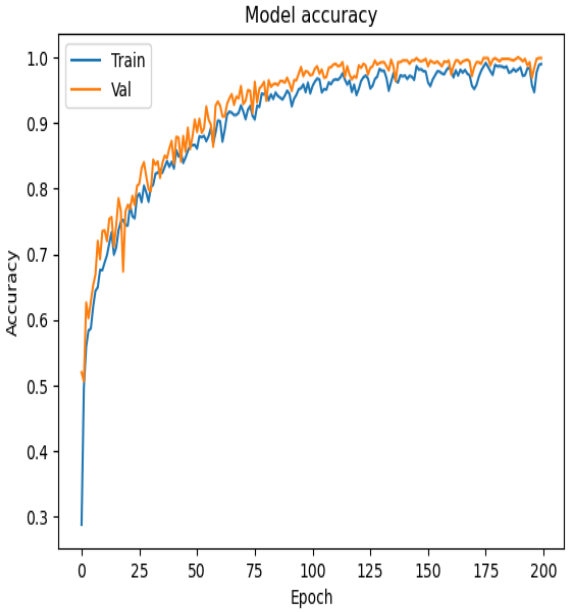


Fig (10) The Accuracy curve of the training and validation curve for the $\mathbf{DCNN_2}$ model.

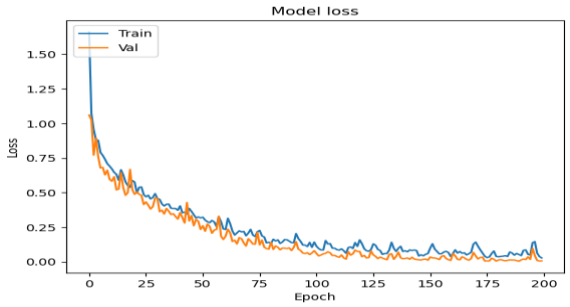


Fig (11) The Loss curve of training and validation curve for the $\mathbf{DCNN_2}$ model.

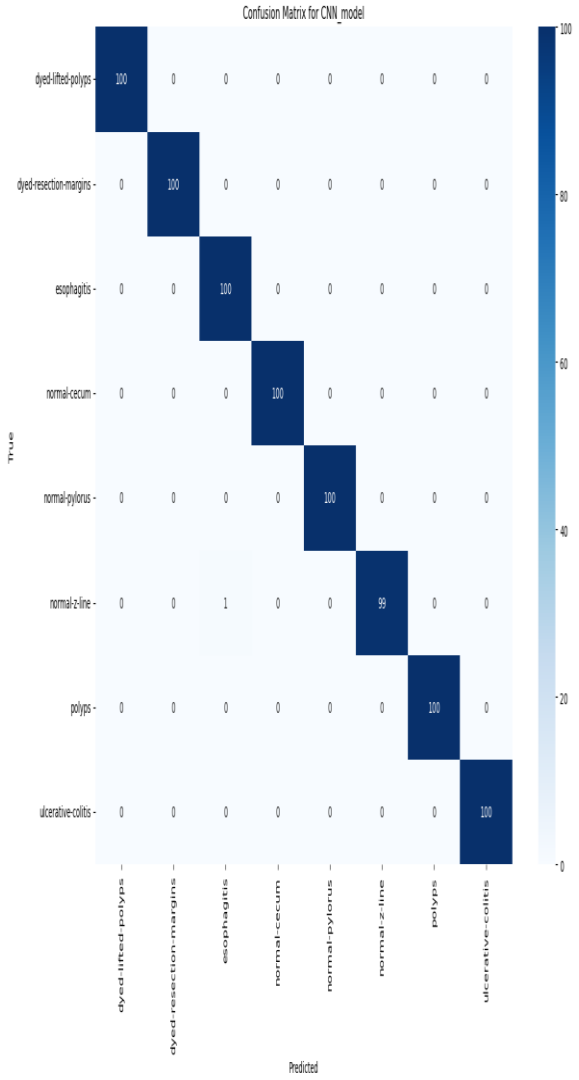


Fig (12) Confusion matrix for $\mathbf{DCNN_2}$ model.

5.3 The 3rd approach based on Ensemble Learning

In this approach, EL is employed to enhance the total performance by deploying two or more models, at least one of all has poor performance, but one or rest others have a very good performance, then the total performance will coverage the poor efficiency of the weak model, so there will one proposed experimental approach in this section by utilizing $DCNN_1$ and $DCNN_2$.

The EL model can be explained in Figure (13), which declares that the two models $\mathbf{(DCNN_1, DCNN_2)}$ are trained together for the same training dataset, then their predictions are tested on the same test dataset, and lastly, EL votes for their best performance. This EL model consumes the total parameters and memory cost of all of them, which are 15443056, and 58.91 MB. After training, the accuracy curve and loss curve are gathered and plotted in Figure (14). As shown, the accuracy for testing is about **0.995**, and the time complexity is 6 hours,23 minutes, and 50 seconds.

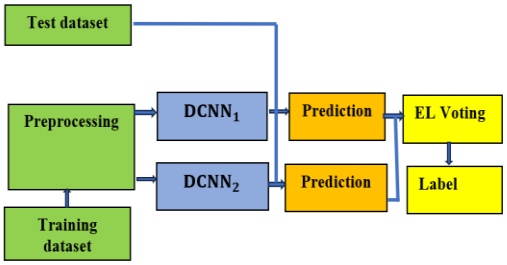


Fig (13) The EL model architecture.

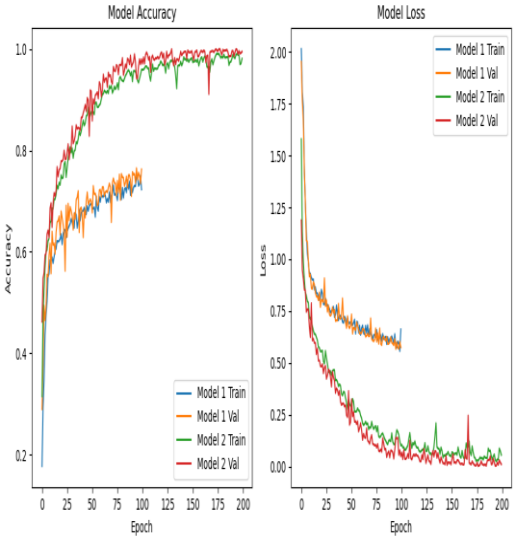


Fig (14) The EL model (**$DCNN_1$,$DCNN_2$**) accuracy curve, and loss curve.

5.3 The 4th approach based on Transfer Learning (VGG-16)

The Visual Geometry Group at Oxford University created VGGs, which have a deep and consistent structure based on 3x3 convolutional filters layered in several layers. Well-known variations such as VGG-16 and VGG-19 demonstrated the power of depth in CNNs by achieving state-of-the-art performance on the ImageNet dataset.

In this approach, based on the TL technique, a pretrained global model, **VGG-16,** will be employed to classify 8 GIDs. The method, as before, depends on resources for processing the model are the same as the first, second, third, and fourth approaches, which were a CPU core i5, RAM 16 gigabytes, and GPU Intel. For software resources, Windows 10 is the operating system, and Python 3.7 is a programming language, written by the Spyder compiler in the Anaconda platform.

The images in the eight classes were resized to 224224 pixels, as was done in the first approach. The dataset was split into two groups for training and testing operations, with a ratio of 80:20 %. This resulted in 3200 images for training and 800 images for validation. Besides, data augmentation was applied in all the next experimental approach (**VGG-16**).

The results in this experimental approach, **VGG-16,** the training twice, including hyperparameters. The batch size for training was equal to 128, the number of epochs was equal to 25 at 1st time and 50 at 2nd time, the optimizer employed was Adam, and the last **VGG-16** fully connected layer was adapted to 8 because the dataset has 8 classes.

Table (3) represents a summary of **VGG-16,** and the total number of hyperparameters is equal to 14915400, occupying 56.90 MB, trainable parameters equal to 200712, occupying 784.03 KB, non-trainable parameters equal to 14714688, occupying 56.13 MB. After training, the accuracy curve, loss curve, and confusion matrix are gathered and plotted in Figures (15), (16), and (17). As shown, the best accuracy for testing the 2nd time was about **0.9800**, and the time complexity is 8 hours, 5 minutes, and 11 seconds.

Table (3) **VGG-16** Summary and total parameters.

The Layer	Layer Output Shape	No of Parameters	
input layer (Input Layer)	(None, 224, 224, 3)	0	
conv2d#1 (Conv2D)	(None, 224, 224, 64)	1,792	
conv2d#2_(Conv2D)	(None, 224, 224, 64)	36,928	
max_pooling2d	(None, 112, 112, 64)	0	
conv2d#3(Conv2D)	(None, 112, 112, 128)	73,856	
conv2d#4(Conv2D)	(None, 112, 112, 128)	147,584	
max_pooling2d	(None, 56, 56, 128)	0	
conv2d#5(Conv2D)	(None, 56, 56, 256)	295,168	
conv2d#6(Conv2D)	(None, 56, 56, 256)	590,080	
conv2d#7(Conv2D)	(None, 56, 56, 256)	590,080	
max_pooling2d	(None, 28, 28, 256)	0	
conv2d#8(Conv2D)	(None, 28, 28, 512)	1,180,160	
conv2d#9(Conv2D)	(None, 28, 28, 512)	2,359,808	
conv2d#10(Conv2D)	(None, 28, 28, 512)	2,359,808	
max_pooling2d	(None, 14, 14, 512)	0	
conv2d#11(Conv2D)	(None, 14, 14, 512)	2,359,808	
conv2d#12(Conv2D)	(None, 14, 14, 512)	2,359,808	
conv2d#13(Conv2D)	(None, 14, 14, 512)	2,359,808	
max_pooling2d	(None, 7, 7, 512)	0	
flatten (Flatten)	(None, 25088)	0	**Trainable parameters**
dense#2 (Dense)	(None, 8)	200,712	

Non trainable parameters

Total parameters =14915400 (56.90MB), {Trainable params: 200712 (784.03 KB)}, {non-trainable params: 14714688 (56.13 MB)}

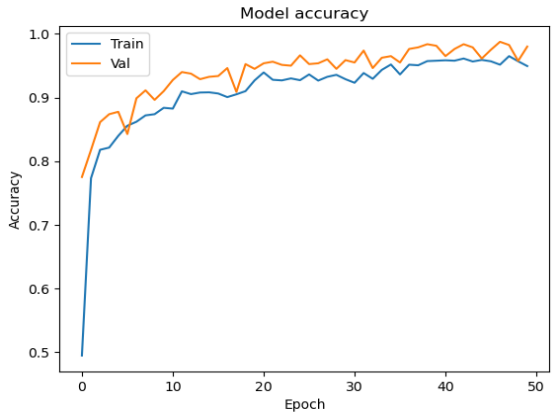


Fig (15) The Accuracy curve of the training and validation curve for the **VGG − 16** model

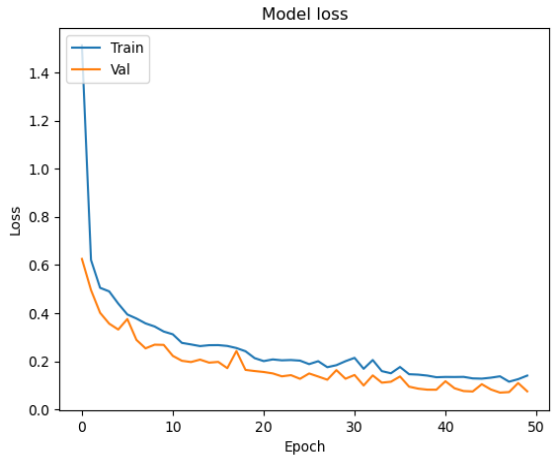


Fig (16) The Loss curve of training and validation curve for the **VGG − 16** model.

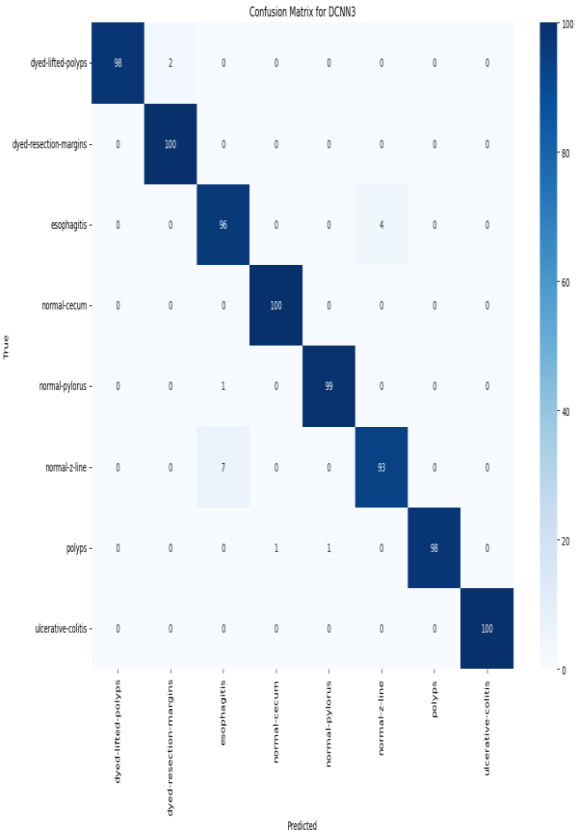


Fig (17) Confusion matrix for **VGG − 16** Model.

5.4 The 4 approaches, analysis and comparison

The 4 approaches were done under different tuning parameters, but all of them used the same resources and implementation tools. From the results, it can be obtained next items.

- the model **$DCNN_1$** The accuracy result for testing is about **0.75125**, under 100 epochs,3 dropouts (0.2), SGD optimizer. This model struggled to classify 4 classes, especially dyed-Resection-Margin, dyed-Lifted-Polyps, Polyps, and esophagitis; even other 4 classes have better performance results, but the Accuracy is still poor, which means that the model's performance is weak, because of using the SGD optimizer, and that will be ensured in the 2nd approach, **DCNN2** by applying Adam optimizer.

- The model **$DCNN_2$** The accuracy results for testing is about **0.99875**, under 200 epochs,1 dropout (0.2), Adam optimizer. This model is deeper than **DCNN1** and has a different optimizer (Adam), which resulted in a better diagnosis than **DCNN1**. This model has achieved super excellent Accuracy; it has only one misclassification to classify 1 class, normal z line, but the other 7 classes have **100%** accuracy results.

- The 3rd approach, which applied EL results, ensured that EL can improve the diagnosis performance of two models, one of which has better diagnosis than the other, by voting that the EL model has achieved the same accuracy as **DCNN2 (0.995).**

- The 4th approach based on TL(**VGG-16**) has achieved very good accuracy, near to the EL approach, but it consumes a lot of time (**0.9800**).

- In the next table (4), A comparison between the 4 approaches for GIDs diagnosis.

Table (4) The 4 approaches results, Summary, and comparison

Approach	Total parameters + memory cost	Hyperparameter +optimizer (Epoch, batch size, dropout, opt)	Results				
			Acc	Spec	Sens	Perc	F1-score
1st Approach ($DCNN_1$)	11169992, (42.61 MB)	100,128,3, SGD	0.75	0.75	0.7	0.75	0.75
2nd Approach ($DCNN_2$)	4273064, (16.30 MB)	200,128,1, Adam	0.99	0.99	0.99	0.99	0.99
3rd Approach (EL model)	15443056, (58.91 MB)	$DCNN_1$, $DCNN_2$	0.99	0.99	0.99	0.99	0.99
4th Approach (TL model)	14915400 (56.90MB)	Pretrained (VGG16),50,128, Adam	0.98	0.98	0.98	0.98	0.98

In the next table (5), A comparison between the 4 approaches And previous works For GIDs diagnosis.

Table (5) The 4 approaches, results and previous works, Summary, and comparison.

Approach	Dat aset	Methods	Results accuracy
M. H. Al-Adhaileh Al-Adhaileh, E. M. Senan, et el[10].	Kvasir dataset	AlexNet, ResNet-101, and Google Net	0.97
T. Agrawal, and C.E. Wilson[17].		The Inception-V3 and VGGNet features	0.961
A. Asperti and C. Mastronardo[12].		TL, CNN, Data Augmentation Techniques, and Snapshot Ensemble	0.915
1st Approach ($DCNN_1$)		3 conv+3 maxpool+2 fully connected	0.75125
2nd Approach ($DCNN_2$)		5 conv+5 maxpool+2 fully connected	0.99875
3rd Approach (EL model)		$DCNN_1, DCNN_2$	0.995
4th Approach (TL model)		VGG-16	0.9800

6. Conclusions and Future Work.

Based on the four experimental methods discussed in the previous part, which used the balanced kavsir dataset 4000 for 8 multi-gastrointestinal diseases, under different hyperparameters, it can be inferred that **DCNN**, as in **DCNN1** and **DCNN2**, may produce classification solving accuracy which ranges from low (**0.75**) to high (**0.995**). As can be seen, **DCNN2** has a deeper network, was set up with Adam rather than SGD, and has more epochs than **DCNN1**, making it a more precise diagnosis.

Additionally, by utilizing **EL**, in which **DCNN1** is ensembled with **DCNN2**, the **EL** model produced high accuracy (**99.5**) and compensated for **DCNN1's** weakness.

Although **TL** produced a high accuracy (**0.98**), which ensured that it can be used with the pretrained model to get a high accuracy in medical diagnosis problems.

The following points will be the main focus of the future work.

- Applying comparable techniques to other multi-class illnesses, which have a low and imbalanced dataset, to examine the outcome and evaluate the advantages and disadvantages.
- Applying comparable techniques to classification problems that are not specifically in the biomedical sector but rather in another domain, like agriculture, where datasets are different, to examine the outcomes and weigh the advantages and disadvantages of EL and TL.

References

[1] Gallo, Antonella, et al, "Main Disorders of Gastrointestinal Tract in Older People: An Overview." Gastrointestinal Disorders 6.1, pp. 313-336,2024.

[2] Naz, Javeria, Muhammad Sharif, Musarrat Yasmin, Mudassar Raza, and Muhammad A. Khan, "Detection and classification of gastrointestinal diseases using machine learning." Current Medical Imaging 17, no. 4, pp. 479-490, 2021.

[3] Borgli, Hanna, et al, "HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy." Scientific data 7.1, p. 283, 2020..

[4] Khan, Asifullah, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi, "A survey of the recent architectures of deep convolutional neural networks." Artificial intelligence review 53, pp. 5455-5516,2020.

[5] Dong, Xibin, et al. "A survey on ensemble learning." Frontiers of Computer Science 14 (2020): 241-258.

[6] Weiss, Karl, Taghi M. Khoshgoftaar, and DingDing Wang. "A survey of transfer learning." Journal of Big data 3 (2016): 1-40.

[7] Pogorelov, Konstantin, et al. "Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection." Proceedings of the 8th ACM on Multimedia Systems Conference, 2017.

[8] Alhajlah, Mousa, et al. "Gastrointestinal diseases classification using deep transfer learning and features optimization." Comput. Mater. Contin 75.1 (2023): 2227-2245.

[9] G. L. H. Wong, A. J. Ma, H. Deng, J. Y. L. Ching and V. W. S. Wong, "Machine learning model to predict recurrent ulcer bleeding in patients with a history of idiopathic gastroduodenal ulcer bleeding," Alimentary Pharmacology & Therapeutics, vol. 49, pp. 912–918, 2019.

[10] M. H. Al-Adhaileh, E. M. Senan, F. W. Alsaade, T. H. H. Aldhyani and N. Alsharif, "Deep learning algorithms for detection and classification of gastrointestinal diseases," Complexity, vol. 2021, p. 29434, 2021.

[11] S. Nadeem, M.Atif Tahir, S.Sadiq Ali Naqvi, and M. Zaid, 2018, "Ensemble of Texture and Deep Learning Features for Finding Abnormalities in the Gastrointestinal Tract", In Proceedings of 10th International Conference on Computational Collective Intelligence, pp. 469–478.

[12] A.. Asperti, and C. Mastronardo1, 2018, "The Effectiveness of Data Augmentation for Detection of Gastrointestinal Diseases from Endoscopical Images", In Proceedings of the 5th International Conference on Bioimaging, pp . 213-220.

[13] I. Guyon, I. Dror, G. Lemaire, V. Taylor, and G. Silver, 2012, "Deep learning of representations for unsupervised and transfer learning", In Proceedings of the 27 Conference on Unsupervised and Transfer Learning Challenges in Machine Learning, pp . 17-37.

[14] L. Cun, Y. Boser, B.Denker, J.S., Henderson, D. Howard,R. E. Hubbard, and W. Jackel, 1989, "Back propagation applied to handwritten zip code recognition", Neural Computation, Vol 1, Issue 4.

[15] S.C. Wong, A.Gatt, V. Stamatescu, and M.D.McDonnell, 2016, "Understanding data augmentation for classification: when to warp? ", In Proceedings of the 2016 International Conference on

Digital Image Computing: Techniques and Applications (DICTA), pp . 82-87.

[16] G. Huang, Y.Li, G.Pleiss, Z. Liu, J.E.Hopcroft, and K.Q.Weinberger, 2017, "Snapshot ensembles: Train 1, get M for free", In Proceedings of 5th International Conference on Learning Representations, pp . 50–64.

[17] T. Agrawal, R. Gupta, S.Sahu, and C.E. Wilson, 2017, "SCL-UMD at the Medico Task-MediaEval 2017: Transfer learning Based Classification of Medical Images", In Proceedings of the MediaEval 2017 Workshop Conference.

[18] Sadhana, S., and R. Mallika, "An intelligent technique for detection of diabetic retinopathy using improved alexnet model based convolutional neural network." Journal of Intelligent & Fuzzy Systems 40.4, pp. 7623-7634,2021.

[19] Ndong, P.S.B.; Adoni, W.Y.H.; Nahhal, T.; Kimpolo, C.; Krichen, M.; Byed, A.E.; Assayad, I.; Mutombo, F.K. A face-mask detection system based on deep learning convolutional neural networks. In Advances on Smart and Soft Computing: Proceedings of ICACIn 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 273–283.

[20] Krichen, Moez. "Convolutional neural networks: A survey." Computers 12.8 (2023): 151.

[21] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of big data 6.1 (2019): 1-48.

[22] Dong, Xibin, et al. "A survey on ensemble learning." Frontiers of Computer Science 14 (2020): 241-258.

[23] Zhou Z H. Ensemble Methods: Foundations and Algorithms. Chapman and Hall/CRC, 2012.

[24] Weiss, Karl, Taghi M. Khoshgoftaar, and DingDing Wang. "A survey of transfer learning." Journal of Big data 3 (2016): 1-40.

[25] Shimodaira H. Improving predictive inference under covariate shift by weighting the log-likelihood function. J StatPlan Inf. 2000;90(2):227–44.

[26] Zhuang, Fuzhen, et al. "A comprehensive survey on transfer learning." Proceedi(2020): 43-76.

[27] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2010;22(10):1345–59.